



图文 41 基于MQ实现订单数据同步给大数据团队，应该如何设计

507 人次阅读

2019-11-29 08:28:08

[详情](#) [评论](#)

基于MQ实现订单数据同步给大数据团队，应该如何设计？

石杉老哥重磅力作：《互联网java工程师面试突击》（第3季）【强烈推荐】：



全程真题驱动，精研Java面试中6大专题的高频考点，从面试官的角度剖析面试

(点击下方蓝字试听)

[《互联网Java工程师面试突击》（第3季）](#)

1、下面应该解决哪个问题？

小猛已经成功解决了订单系统的两个问题，接着开始思考下一个应该解决的问题。小猛对着订单系统的问题列表陷入了思考：

下单核心流程环节太多，性能较差

订单退款的流程可能面临退款失败的风险

关闭过期订单的时候，存在扫描大量订单数据的问题

跟第三方系统耦合在一起，性能存在抖动的风险
大数据团队要获取订单数据，存在不规范直接查询订单数据库的问题
做秒杀活动时订单数据库压力过大

目前第一个和第四个问题都已经解决掉了，接下来哪个问题是比较紧急需要解决的呢？

小猛想了一下，觉得订单退款失败并不是太紧急的问题，因为这个问题偶尔才会出现

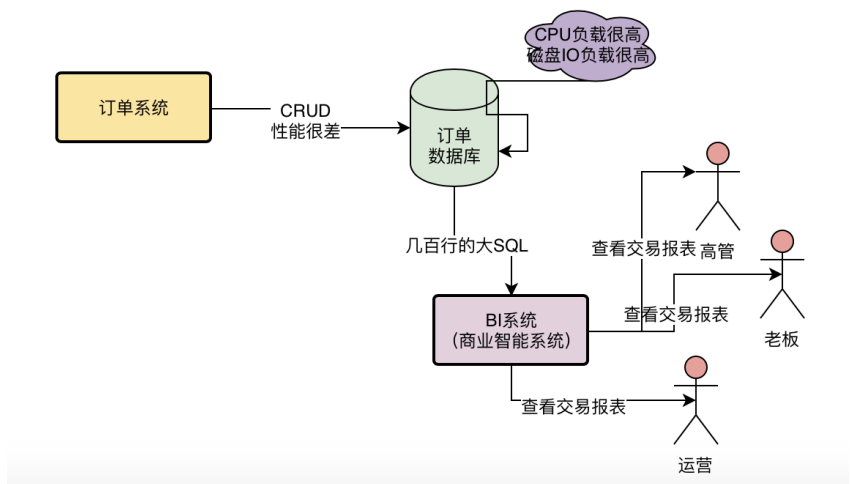
另外订单数据量也不是特别大，因此关闭订单时扫描过多订单的问题也不是特别紧急。

这个时候，小猛认为大数据团队每天都要在订单数据库上执行上百次几百行的复杂大SQL，有时候会导致订单数据库性能抖动这个问题，是当前应该要解决的。

2、大数据团队的几百行大SQL是如何影响订单数据库的？

小猛想到这里，先回忆了一下大数据团队的几百行大SQL是如何影响订单数据库的

小猛在纸上迅速的画出了下面的一幅图。



在图里很清晰地看到，大数据团队的BI系统每天都会直接在订单数据库里执行上百次几百行的大SQL，而每次一个几百行大SQL的执行都需要耗时几秒到十几秒不等

每次这样一个几百行的大SQL执行，都会导致MySQL数据库服务器的资源负载急剧抖动，会让CPU、内存、磁盘IO的负载都瞬间升高。

而一旦MySQL数据库的资源负载瞬间升高，会导致订单系统在MySQL数据库上执行的SQL语句性能出现急剧下降，因此会导致订单系统的性能也出现抖动。

这其实就是大数据团队目前对订单系统的影响。

3、如何避免大数据团队直接查询订单数据库？

小猛接着继续思考，如何解决这个问题呢？

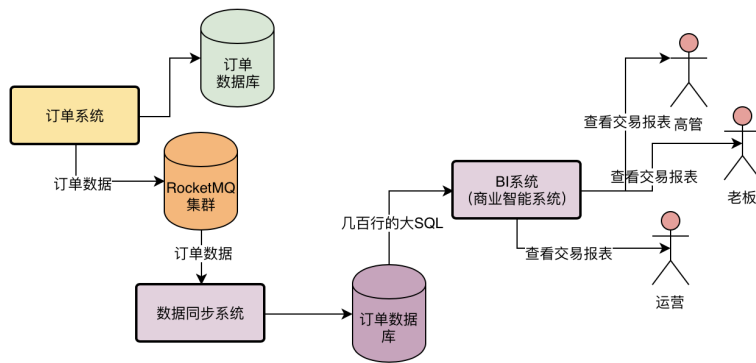
实际上要解决这个问题，就必须避免大数据团队直接查询订单数据库

那么又如何避免大数据团队直接查询订单数据库呢？这是一个最关键的问题。

其实很简单，完全可以由订单系统将订单数据推送到一个MQ里，然后大数据团队从MQ里获取订单数据，接着将订单数据落地到自己的存储中去

比如最简单的办法，就是将订单数据落地到大数据团队自己的一个MySQL数据库中，然后从自己的MySQL数据库里统计报表。

小猛一边思考，一边在纸上刷刷的画出了下面这个草图。



如果能够采用这样的一个架构，那么大数据团队的几百行的大SQL就不会对订单数据库造成任何影响了。

4、TopicOrderPaySuccess里的订单支付成功消息可以使用吗？

下一个问题，订单系统应该如何将订单数据发送到RocketMQ里去呢？

之前我们讲过，已经让订单系统在支付成功的时候，将订单支付成功的消息发送到了RocketMQ里去了，然后其他系统会订阅这个订单支付成功的消息去进行对应的业务处理。

但是这个订单支付成功的消息，足够让大数据团队使用吗？

显然是不行的，因为大数据团队需要的是跟订单数据库一模一样的一份完整的数据，而不仅仅是订单支付成功的消息，所以不能直接使用之前“TopicOrderPaySuccess”这个Topic里的消息。

因此，我们还是需要想办法将完整的订单数据都发送到RocketMQ里去，然后让大数据团队去获取。

5、如何将完整的订单数据发送到RocketMQ里去？

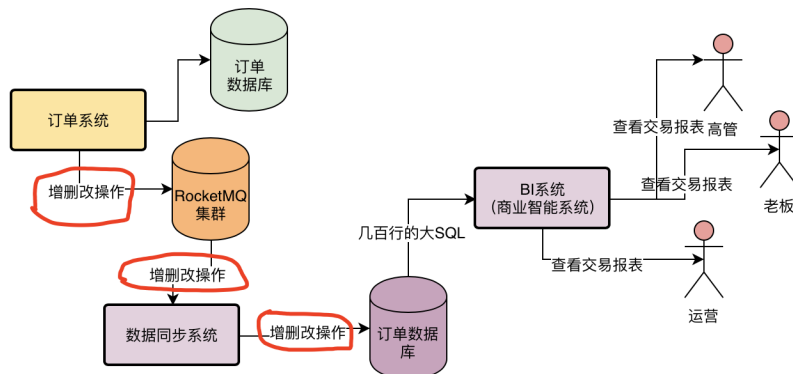
这时小猛开始思考如何将完整的订单数据发送到RocketMQ里去

实际上一个比较简单的办法，就是在订单系统中但凡对订单执行增删改类的操作，就把这种对订单增删改的操作发送到RocketMQ里去。

然后大数据团队的数据同步系统从RocketMQ里获取到订单增删改的操作，就会在自己的数据库里执行一样的增删改操作。

通过还原执行一样的insert、update和delete语句，就可以在自己的数据库里还原出来一样的订单数据。

小猛边想边画出了下面的草图。



但是这种方案的一个问题就是订单系统为了将数据同步给大数据团队，必须在自己的代码里耦合大量的代码去发送增删改操作到RocketMQ，这会导致订单系统的代码出现严重的污染，因为这些发送增删改操作到RocketMQ里的代码是跟订单业务没关系的。

小猛灵机一动，想到了另外一个好办法。前段时间他听了其他团队的一个技术分享，目前公司已经部署了一种MySQL Binlog同步系统

这种系统会监听MySQL数据库的Binlog，所谓Binlog大致可以理解为MySQL的增删改操作日志。

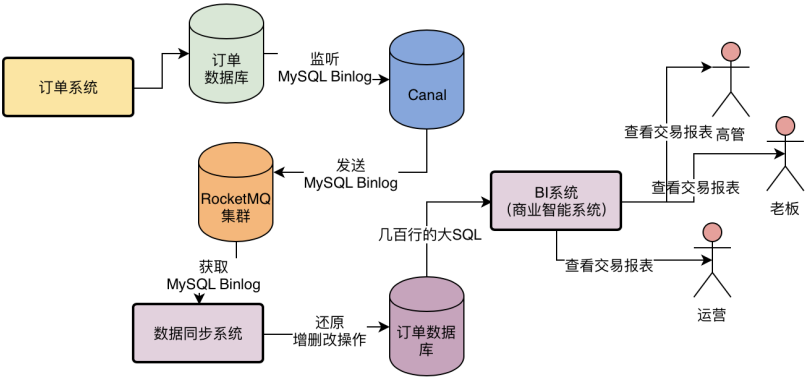
然后MySQL Binlog同步系统会将监听到的MySQL Binlog（也就是增删改操作日志）发送给你的系统，让你来处理这些增删改操作日志。

这种MySQL Binlog系统现在是有不少成熟的开源技术方案的，比如阿里开源的Canal，以及Linkedin开源的Databus，都可以监听MySQL Binlog，然后将MySQL Binlog发送给你的系统，交给你去处理。

因此完全可以将数据同步方案修改为如下所示，采用Canal监听MySQL Binlog，然后直接发送到RocketMQ里

然后大数据团队的数据同步系统从RocketMQ中获取到MySQL Binlog，也就获取到了订单数据库的增删改操作，接着把增删改操作还原到自己的数据库中去就可以。

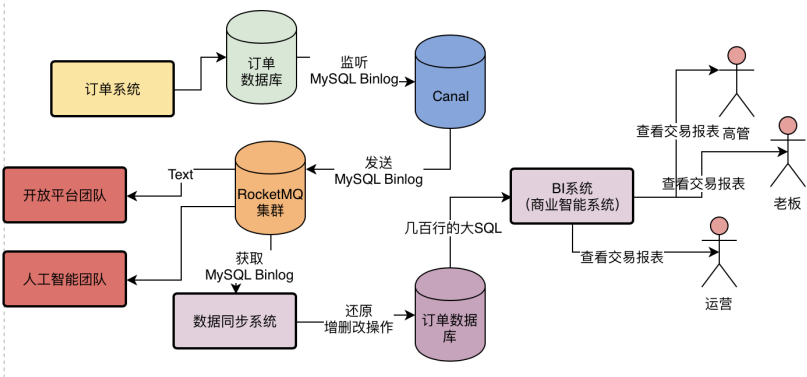
小猛想着就画出了下面的草图。



而且这样的一套方案还有一个额外的好处，就是由订单技术团队将完整的订单数据库的MySQL Binlog推送到RocketMQ里

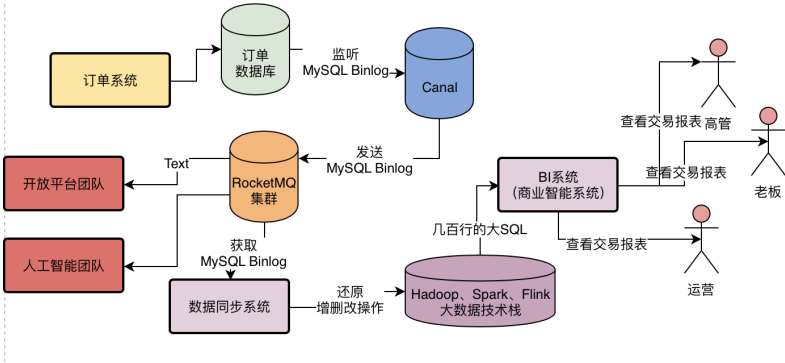
无论是大数据团队，还是未来公司的其他技术团队，比如说开放平台团队，人工智能团队，等等，只要想要订单数据，都可以直接从这个RocketMQ里去获取完整的订单数据。

小猛想着想着，在草图中继续加入了一些东西。



说到这里，我们要给大家解释一下，实际上大数据团队并没有必要仅仅只通过MySQL来出数据报表，完全可以采用Hadoop、Spark、Flink等大数据技术来出数据报表。

因此实际上看起来大数据团队那部分的图应该是如下图所示的。



6、一点小小的总结

今天的内容讲完了，我们来做一点小小的总结，我们今天分析了目前大数据团队对订单数据库造成的压力以及原因，接着分析了解决这个问题的方案，就是将订单数据同步给大数据团队，让他们自己处理。

然后我们一步步分析了如何将数据同步给大数据团队，先是考虑在订单系统代码内部嵌入一些额外的代码，将订单的增删改操作发送到RocketMQ里，但是后来发现这样会导致污染订单系统的代码。

所以后来我们提出了一个完美的解决方案，就是用Canal、Databus这样的MySQL Binlog同步系统，监听订单数据库的binlog发送到RocketMQ里

然后大数据团队的数据同步系统从RocketMQ里获取订单数据的增删改binlog日志，还原到自己的数据存储中去，可以是自己的数据库，或者是Hadoop之类的大数据生态技术。

然后大数据团队将完整的订单数据还原到自己的数据存储中，就可以根据自己的技术能力去出数据报表了，不会再影响订单系统的数据库了。

到这里为止，小猛发现自己通过MQ技术又解决掉了订单系统的一个问题，他列出了下面的问题列表：

- 下单核心流程环节太多，性能较差
- 订单退款的流程可能面临退款失败的风险
- 关闭过期订单的时候，存在扫描大量订单数据的问题
- 跟第三方系统耦合在一起，性能存在抖动的风险
- 大数据团队要获取订单数据，存在不规范直接查询订单数据库的问题
- 做秒杀活动时订单数据库压力过大

End

专栏版权归公众号**狸猫技术窝**所有

未经许可不得传播，如有侵权将追究法律责任

狸猫技术窝其他精品专栏推荐：

- [《从零开始带你成为JVM实战高手》](#)
- [《21天Java 面试突击训练营》（分布式篇）](#)（现更名为：[互联网Java工程师面试突击第2季](#)）
- [互联网Java工程师面试突击（第1季）](#)

重要说明：

如何提问：每篇文章都有评论区，大家可以尽情在评论区留言提问，我会逐一答疑

如何加群：购买了狸猫技术窝专栏的小伙伴都可以加入**狸猫技术交流群**

具体加群方式，请参见**目录菜单**下的文档：《付费用户如何加群？》（**购买后可见**）